UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/799,336	03/12/2004	Gary S. Henderson	13768.1160	6236

47973 7590 03/19/2009

WORKMAN NYDEGGER/MICROSOFT
1000 EAGLE GATE TOWER
60 EAST SOUTH TEMPLE
SALT LAKE CITY, UT 84111

EXAMINER	
LE, MIRANDA	
ART UNIT	**PAPER NUMBER**
2169	

MAIL DATE	DELIVERY MODE
03/19/2009	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/799,336	HENDERSON ET AL.	
	Examiner	Art Unit	
	MIRANDA LE	2169	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *15 January 2009*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-20* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-20* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date *01/07/09, 12/15/08*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Continued Examination Under 37 CFR 1.114

A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 01/15/2009 has been entered.

This communication is responsive to Amendment, filed 01/15/09.

Claims 1-20 are pending in this application. This action is made non-Final.

Information Disclosure Statement

Applicants' Information Disclosure Statements, filed 01/07/09, 12/15/08, have been received, entered into the record, and considered. See attached form PTO-1449.

Claim Objections

Claim14 is objected to because of the following informalities: Claim 14, step (ii), "(b) prerequisites *indentifying*" should be changed to "(b) prerequisites *identifying*".

Appropriate correction is required.

Claim 6 is objected to because of the following informalities: Claim 6, line 8, “having stored *there* computer executable instruction" should have changed to “having stored *thereon* computer executable instruction".

Appropriate correction is required.

Claim Rejections - 35 USC § 112

The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

Claim 11 is rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Claim 11 recites the limitation "the client computer" in the last line of claim 11. There is insufficient antecedent basis for this limitation in the claim.

Claim Rejections - 35 USC § 101

35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

Claims 6- 10 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

Claim 6 recites “A computer program product comprising a computer-readable medium...”, however, the claim fails to place the invention squarely within one statutory class of invention. There exists no explanatory or defining

language in the specification or elsewhere in the claims to enable the Examiner to determine which media, in particular, Applicant seeks to include in this claim. It is a reasonable interpretation of this claim language that Applicant may be attempting to include computer readable media that falls outside the limits of § 101, for example, transmission media. In paragraph [0013] of the instant specification, Applicant has provided evidences that Applicant intends the "medium" to include signals such as "transmission media". Further, the term bearing implies that the computer readable medium is a carrier wave, thereby the machine readable medium is an intangible embodiment.

The claim is drawn to a form of energy. Energy is not one of the four categories of invention and therefore this claim(s) is/are not statutory. Energy is not a series of steps or acts and thus is not a process. Energy is not a physical article or object and as such is not a machine or manufacture. Energy is not a combination of substances and therefor not a composition of matter.

A computer-readable medium including a carrier wave, or signal, is non-statutory subject matter as set forth in MPEP 2106 (IV)(B)(2)(a). Therefore, claim 19 is not limited to tangible embodiments, instead being sufficiently broad so as to encompass intangible media such as transmission media; the claims are not limited to statutory subject matter and are therefore non-statutory.

Applicant is advised to amend the claims as "A computer product comprising a computer readable *storage* medium…"; also amend the specification to include the term "computer readable *storage* medium"; in order to overcome the 101 issues.

Claims 7-10 incorporate the deficiencies of claim 6, respectively, and do not add tangibility to the claimed subject matter, they are likewise rejected.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

Claims 1-3, 5-8, 10, 14-16 are rejected under 35 U.S.C. 103(a) as being unpatentable over Gazdik et al. (US Patent No. 6,301,708), in view of Lee, Kyu-Woong et al. (US Public No. 20040031029).

As to claims 1, 6, Gazdik teaches at a computing device having a normal installation behavior for use during a software update installation, the normal installation behavior including a normal user-interface behavior and a normal download behavior used when installing a standard software update, the normal user-interface behavior including presenting a description of a software update and a user-interface control for selecting installation of the software update if desired, the normal download behavior including downloading software updates in the background to minimize the impact on network bandwidth, a computer implemented method for modifying the normal installation behaviors of the computing device during a software update installation, the computer implemented method comprising:

obtaining software update information *(i.e. multiple independent files, Summary)* to be installed on the computing device *(i.e. This invention is a method for installing and uninstalling software which fragments the process so that the installation or uninstallation of each component of a software suite is controlled by multiple independent files, Summary)*, the software update information comprising a software update, a rule for applicability of the software update *(i.e. Each component-specific data file contains not only the installation criteria that is used to determine whether or not a software component is installable on a particular computer system, and how to optimize installation of the component on a particular system, but also the command sequences which must be executed for installation and unexecuted during uninstallation, Summary)*, and an installation attribute, the installation attribute indicating that

normal installation behavior at the computing device is to be modified *(i.e. the installer-developer can change the installation flow, Summary)* for installation of the software update *(i.e. the characteristics of individual software components can be modified, Summary)*, the modified installation behavior indicating a modification to one of the normal user-interface behavior and the normal download behavior used when installing a standard update *(i.e. In addition to being able to modify the install flow or add new or modified software components at run time by downloading a new state file or a new component persistent data file from the remote server, the installer engine itself can be updated as well, by simply downloading a new installer engine version from the remote server over the Internet. Only the new and updated files need be downloaded from the remote server, thereby minimizing the amount of information that must be downloaded from the remote server, col. 4, lines 40-58)*; and

determining that the software update is applicable to the computing device based on the rule for applicability and in response to the determination:

modifying the normal installation behavior at the computing device according to the installation attribute to modify at least one of the normal user-interface behavior and normal download behavior used when installing a standard update *(i.e. Another member of the P Command object allows the install program developer to specify whether a command is meant for both normal installation and uninstallation processes, or is exclusive to the install process, or to the uninstall process. For a normal install process, the command ::Execute() method is called; for a normal uninstall process the command ::Un*

Execute() method is called. The other values provide the installer/developer with the ability to limit or refine the functionality of an uninstall or install process, by removing a call to ::Execute() during uninstallation and to ::UnExecute() during installation. This adds a second level of logic encoding into the persistent command objects, which doesn't require the installe/developer to write any additional code or script logic to handle these special conditions, col. 6, lines 50-64); and

installing the software update on the computing device according to the modified installation behavior *(i.e. A component data file may reside at any accessible location, which makes integrated installation from a remote server accessible over the Internet feasible. Component data files that were not in existence at the time the original software distribution package was created can be supplied with new or updated software components via the Internet so that those components can be integrated into an existing software suite at installation run time, Summary)*.

Gazdik implicitly teaches “minimize the impact on network bandwidth” as minimizing the amount of information that must be downloaded from the remote server, col. 4, lines 40-58.

Lee, however, specifically teaches minimize the impact on network bandwidth *(i.e. the distributed bandwidth in the network links and routers to allow the different networked device to more rapidly obtain their own required update files, [0028])*.

It would have been obvious to one of ordinary skill of the art having the teaching of Gazdik, and Lee at the time the invention was made to modify the system of Gazdik to include the limitations as taught by Lee. One of ordinary skill in the art would be motivated to make this combination in order to specify the time when an update for a particular software component in a particular networked device should be performed in view of Lee ([0022), as doing so would give the added benefit of having the distributed bandwidth in the network links and routers to allow the different networked device to more rapidly obtain their own required update files as taught by Lee ([0028]).

As to claims 2, 7, Gazdik, as combined, teaches the installation attribute comprises a mandatory installation attribute *(i.e. A component data file may reside at any accessible location, which makes integrated installation from a remote server accessible over the Internet feasible. Component data files that were not in existence at the time the original software distribution package was created can be supplied with new or updated software components via the Internet so that those components can be integrated into an existing software suite at installation run time, Summary)*; and

wherein modifying the normal installation behavior of the computing device according to the mandatory installation attribute comprises requiring the software update to be installed on the computing device and providing a visual indication in the user-interface that a user is unable to unselect installation of the software update *(i.e. The functionality of a state object is defined by its*

::ShowState() method, which the state machine calls and executes. A new state can be added as a derivative of the PState object. By overriding the ::ShowState() method. This new state can provide any functionality that is needed for an install or uninstall process. The architecture provided by the present invention facilitates the addition of new install/uninstall state functionality, col. 4, line 66 to col. 5, line 23).

As to claims 3, 8, Lee, as combined, teaches the installation attribute comprises a priority installation attribute; and

wherein modifying the normal installation behavior of the computing device according to the priority installation attribute comprises modifying the standard download behavior to permit the download process downloading the software update to compete with other network activities on a current connection so that as much network bandwidth as possible is used when downloading the software update's content so as to download the software update more quickly over the current connection *(i.e. As another example, if the installation tasks were handled from a central location (e.g., from the centralized software update engine), the processing burden associated with processing a large number of updates using a single processing engine would have degraded performance. In contrast, the invention takes advantage of the distributed bandwidth in the network links and routers to allow the different networked device to more rapidly obtain their own required update files. Furthermore, the invention takes*

advantage of the distributed processing power in the networked devices to share the processing burden associated with updates, [0028]).

As to claims 5, 10, Lee, as combined, teaches the installation attribute comprises a zero interruption (ZSI) installation attribute, the ZSI attribute indicating that the software update is a ZSI software update configured so as to not cause any software interruptions during installation *(i.e. The update schedule specifies the time when an update for a particular software component in a particular networked device should be performed. Optionally, the update schedule may also include a priority classification for the update. When the scheduled time arrives to update a particular software component on a particular networked device, a software update engine (which may include one or more individual sub-engines) sends the update parameters regarding the update file, along with any other parameters relevant to the update, to a local update agent local to the particular networked device on which the software component to be updated is located. The information sent includes, for example, parameters indicating where in the network or on the Internet the actual update file may be found and downloaded, [0022])*; and

wherein modifying the normal installation behavior of the computing device according to the ZSI installation attribute comprises:

determining that the computing device is configured for automatic installation of ZSI software updates, and in response to the determination, automatically installing the software update without further user interaction *(i.e.*

FIGS. 7A and 7B are flowcharts illustrating, in accordance with one embodiment of the present invention, the steps involved in automatically updating network software components, [0018]).

As per claim 14, Lee, as combined, teaches:

synchronizing available updates with a software update service, including traversing a software update hierarchy to identify specific software updates that apply to the computing device, the software update hierarchy including at least a base set of one or more base updates and a second set of one or more second tier updates *(i.e. The update schedule specifies the time when an update for a particular software component in a particular networked device should be performed. Optionally, the update schedule may also include a priority classification for the update. When the scheduled time arrives to update a particular software component on a particular networked device, a software update engine (which may include one or more individual sub-engines) sends the update parameters regarding the update file, along with any other parameters relevant to the update, to a local update agent local to the particular networked device on which the software component to be updated is located. The information sent includes, for example, parameters indicating where in the network or on the Internet the actual update file may be found and downloaded, [0022])*, wherein

each base update includes one or more applicability condition to be tested at the computing device to determine if the base update is applicable to the

computing device *(i.e. There is also included a software update engine configured to send individual sets of the update parameters to individual ones of the plurality of local update agents at the plurality of network devices, wherein a first one of the plurality of local update agents disposed at a first one of the plurality of networked devices is configured to obtain, upon receiving a first one of the individual sets of the update parameters, a first update file using the first one of the individual sets of update parameters. The first update file represents a file containing data for updating a first one of the plurality of software components disposed at the first one of the plurality of networked devices, [0010])*; and

each second tier update includes: (a) one or more applicability conditions to be tested at the computing device to determine if the second tier update is applicable to the computing device, (b) prerequisites identifying one or more base updates that are required for proper installation of the second tier update, and (c) an indicator of whether or not the second tier update is a leaf update *(i.e. For each software component, the update parameters database may include the identification of the specific network device or network devices in which the software component may be found, as well as the current version number of the software component. Furthermore, the information in the update parameters database may also include information pertaining to any update file for that software component, [0020])*.

As per claim 15, Lee, as combined, teaches obtaining software update information to be installed on the computing device comprises receiving a

selection of one or more software updates form among the identified specific software updates that apply to the computing device *(i.e. As another example, if the installation tasks were handled from a central location (e.g., from the centralized software update engine), the processing burden associated with processing a large number of updates using a single processing engine would have degraded performance. In contrast, the invention takes advantage of the distributed bandwidth in the network links and routers to allow the different networked device to more rapidly obtain their own required update files. Furthermore, the invention takes advantage of the distributed processing power in the networked devices to share the processing burden associated with updates, [0028])*.

As per claim 16, Lee, as combined, teaches modifying the normal installation behavior at the computing device according to the installation attribute comprises modifying a background download process that normally determines the computing device's network usage and makes download requests for software updates based on available remaining bandwidth *(i.e. As another example, if the installation tasks were handled from a central location (e.g., from the centralized software update engine), the processing burden associated with processing a large number of updates using a single processing engine would have degraded performance. In contrast, the invention takes advantage of the distributed bandwidth in the network links and routers to allow the different networked device to more rapidly obtain their own required update files.*

Furthermore, the invention takes advantage of the distributed processing power in the networked devices to share the processing burden associated with updates, [0028]).

Claims 4, 9, 11-13, 17-20 are rejected under 35 U.S.C. 103(a) as being unpatentable over Gazdik et al. (US Patent No. 6,301,708), in view of Lee, Kyu-Woong et al. (US Public No. 20040031029), and further in view of Colvin (US Patent No. 6,799,277).

As per claim 11, Gazdik teaches at a computing device having a normal installation behavior for use during a software update installation, the normal installation behavior including a normal user-interface behavior and a normal download behavior used when installing a standard software update, the normal user-interface behavior including presenting a description of software update and a user-interface control for selecting installation of the software update if desired *(i.e. it may ask the user to identify those software components which he desired to install, col. 4, line 66 to col. 5, line 23)*, the normal download behavior including downloading software updates in the background to minimize the impact on network bandwidth *(i.e. minimizing the amount of information that must be downloaded from the remote server, col. 4, lines 40-58)*, a computer implemented method for modifying the normal installation behaviors *(i.e. the installer-developer can change the installation flow, Summary)* of the computing device during a software update installation according to associated installation attributes, the computer implemented method comprising the steps of:

obtaining software update information to be installed on the computing device *(i.e. This invention is a method for installing and uninstalling software which fragments the process so that the installation or uninstallation of each component of a software suite is controlled by multiple independent files, Summary)*, the software update information comprising a software update, a rule for the applicability of the software update, and an installation attribute operable for controlling the installation of the software update *(i.e. Each component-specific data file contains not only the installation criteria that is used to determine whether or not a software component is installable on a particular computer system, and how to optimize installation of the component on a particular system, but also the command sequences which must be executed for installation and unexecuted during uninstallation, Summary)*;

determining whether the installation attribute is a mandatory installation attribute *(i.e. A component data file may reside at any accessible location, which makes integrated installation from a remote server accessible over the Internet feasible. Component data files that were not in existence at the time the original software distribution package was created can be supplied with new or updated software components via the Internet so that those components can be integrated into an existing software suite at installation run time, Summary)*, and if so, requiring the software update to be installed and modifying the standard user-interface behavior of the computing device to provide a visual indication in the user-interface that the user is unable to unselect installation of the software update *(i.e. The state may be displayed to the end user if that type of*

functionality is specified by the state object. Each state object defines a transition table in the state file, which the state machine reads and processes to produce a return code. The state machine transitions from that state object to another by evaluating the return code. Each state object has a specific task. For example, it may prompt the end user to provide a directory name and location; it may ask the user to identify those software components which he desired to install; or it may make one of many other install/uninstall-related queries. The functionality of a state object is defined by its ::ShowState() method, which the state machine calls and executes. A new state can be added as a derivative of the PState object. By overriding the ::ShowState() method. This new state can provide any functionality that is needed for an install or uninstall process. The architecture provided by the present invention facilitates the addition of new install/uninstall state functionality, col. 4, line 66 to col. 5, line 23);

requiring the software update to be installed and modifying the standard user-interface behavior of the computing device to provide a visual indication *(i.e. By overriding the ::ShowState() method. This new state can provide any functionality that is needed for an install or uninstall process. The architecture provided by the present invention facilitates the addition of new install/uninstall state functionality, col. 4, line 66 to col. 5, line 23)* in the user-interface that the user is unable to unselect installation of the software update *(i.e. or it may make one of many other install/uninstall-related queries. The functionality of a state object is defined by its ::ShowState() method, which the state machine calls and executes. A new state can be added as a derivative of the PState object. By*

overriding the ::ShowState() method. This new state can provide any functionality that is needed for an install or uninstall process. The architecture provided by the present invention facilitates the addition of new install/uninstall state functionality, col. 4, line 66 to col. 5, line 23);

determining whether the installation attribute *(i.e. The Priority member thus provides the developer with the ability to apply some sequencing logic to the install and uninstall processes, col. 6, lines 36-49)* is a zero service interruption (ZSI) installation attribute, and if so, modifying the normal installation behavior *(i.e. refine the functionality of an uninstall or install process, col. 6, lines 60-64)* of the computing device with respect to the software update such that the software update will be automatically installed on the computing device without user interaction if the the computing device is properly configured *(i.e. Another member of the P Command object allows the install program developer to specify whether a command is meant for both normal installation and uninstallation processes, or is exclusive to the install process, or to the uninstall process. For a normal install process, the command ::Execute() method is called; for a normal uninstall process the command ::Un Execute() method is called. The other values provide the installer/developer with the ability to limit or refine the functionality of an uninstall or install process, by removing a call to ::Execute() during uninstallation and to ::UnExecute() during installation. This adds a second level of logic encoding into the persistent command objects, which doesn't require the installe/developer to write any additional code or script logic to handle these special conditions, col. 6, lines 50-64)*; and

installing the update on the client computer according to the modified installation behavior *(i.e. A component data file may reside at any accessible location, which makes integrated installation from a remote server accessible over the Internet feasible. Component data files that were not in existence at the time the original software distribution package was created can be supplied with new or updated software components via the Internet so that those components can be integrated into an existing software suite at installation run time, Summary)*.

Gazdik implicitly teaches "minimize the impact on network bandwidth" as minimizing the amount of information that must be downloaded from the remote server, col. 4, lines 40-58, but does not expressly state the term "bandwidth".

Gazdik does not explicitly teach:

determining whether the installation attribute is a deadline installation attribute, and if so, determining whether the corresponding deadline of the deadline attribute has expired.

Lee specifically teaches:

minimize the impact on network bandwidth *(i.e. the distributed bandwidth in the network links and routers to allow the different networked device to more rapidly obtain their own required update files, [0028])*.

The step of determine whether the installation attribute is a deadline installation attribute *(i.e. The update schedule specifies the time when an update for a particular software component in a particular networked device should be performed. Optionally, the update schedule may also include a priority*

classification for the update. When the scheduled time arrives to update a particular software component on a particular networked device, a software update engine (which may include one or more individual sub-engines) sends the update parameters regarding the update file, along with any other parameters relevant to the update, to a local update agent local to the particular networked device on which the software component to be updated is located. The information sent includes, for example, parameters indicating where in the network or on the Internet the actual update file may be found and downloaded, [0022]).

It would have been obvious to one of ordinary skill of the art having the teaching of Gazdik, and Lee at the time the invention was made to modify the system of Gazdik to include the limitations as taught by Lee. One of ordinary skill in the art would be motivated to make this combination in order to specify the time when an update for a particular software component in a particular networked device should be performed in view of Lee ([0022), as doing so would give the added benefit of having the distributed bandwidth in the network links and routers to allow the different networked device to more rapidly obtain their own required update files as taught by Lee ([0028]).

Gazdik, Lee do not specifically teach the step of determining whether the corresponding deadline of the deadline attribute has expired.

Colvin teaches determining whether the corresponding deadline of the deadline attribute has expired *(i.e. If the current authorization interval has expired as indicated at 386, an attempt is made to connect to the server of the*

administrator as indicated at 388. If the connection is successful as determined by block 390, information including a licensed file update with a new authorization interval and/or various other information may be downloaded to the user or user computer as indicated at 396. Otherwise, an error message is displayed as indicated at 392 and the process terminates as represented by block 394, col. 16, lines 23-32).

It would have been obvious to one of ordinary skill of the art having the teaching of Gazdik, Lee, and Colvin at the time the invention was made to modify the system of Gazdik, Lee to include the limitations as taught by Colvin. One of ordinary skill in the art would be motivated to make this combination in order to download a licensed file update with a new authorization interval in view of Colvin (col. 16, lines 23-32), as doing so would give the added benefit of monitoring computer software installed on a plurality of computers in communication with one another or a central computer to form a computer network include associating an activation code or password with the computer software for authorizing one or more copies of the software to be installed on computers associated with the computer network as taught by Colvin (col. 3, lines 21-39).

As to claims 4, 9, Lee, as combined, teaches the installation attribute comprises a deadline installation attribute; and

wherein modifying the normal installation behavior of the computing device according to the deadline installation attribute comprises *(i.e. The update schedule specifies the time when an update for a particular software component*

in a particular networked device should be performed. Optionally, the update schedule may also include a priority classification for the update. When the scheduled time arrives to update a particular software component on a particular networked device, a software update engine (which may include one or more individual sub-engines) sends the update parameters regarding the update file, along with any other parameters relevant to the update, to a local update agent local to the particular networked device on which the software component to be updated is located. The information sent includes, for example, parameters indicating where in the network or on the Internet the actual update file may be found and downloaded, [0022]);

requiring the software update to be installed on the computing device and modifying the standard user-interface behavior to automatically install the software update without further user interaction *(i.e. As another example, if the installation tasks were handled from a central location (e.g., from the centralized software update engine), the processing burden associated with processing a large number of updates using a single processing engine would have degraded performance. In contrast, the invention takes advantage of the distributed bandwidth in the network links and routers to allow the different networked device to more rapidly obtain their own required update files. Furthermore, the invention takes advantage of the distributed processing power in the networked devices to share the processing burden associated with updates, [0028])*.

Gazdik, Lee do not explicitly teach the step of determining that the deadline for the deadline installation attribute has passed.

Colvin teaches the step of determining that the deadline for the deadline installation attribute has passed *(i.e. If the current authorization interval has expired as indicated at 386, an attempt is made to connect to the server of the administrator as indicated at 388. If the connection is successful as determined by block 390, information including a licensed file update with a new authorization interval and/or various other information may be downloaded to the user or user computer as indicated at 396. Otherwise, an error message is displayed as indicated at 392 and the process terminates as represented by block 394, col. 16, lines 23-32)*.

It would have been obvious to one of ordinary skill of the art having the teaching of Gazdik, Lee, and Colvin at the time the invention was made to modify the system of Gazdik, Lee to include the limitations as taught by Colvin. One of ordinary skill in the art would be motivated to make this combination in order to download a licensed file update with a new authorization interval in view of Colvin (col. 16, lines 23-32), as doing so would give the added benefit of monitoring computer software installed on a plurality of computers in communication with one another or a central computer to form a computer network include associating an activation code or password with the computer software for authorizing one or more copies of the software to be installed on computers associated with the computer network as taught by Colvin (col. 3, lines 21-39).

As per claim 12, Lee, as combined, teaches:

sending a configuration request message to the software update service *(i.e. The update schedule specifies the time when an update for a particular software component in a particular networked device should be performed. Optionally, the update schedule may also include a priority classification for the update. When the scheduled time arrives to update a particular software component on a particular networked device, a software update engine (which may include one or more individual sub-engines) sends the update parameters regarding the update file, along with any other parameters relevant to the update, to a local update agent local to the particular networked device on which the software component to be updated is located. The information sent includes, for example, parameters indicating where in the network or on the Internet the actual update file may be found and downloaded, [0022])*;

receiving a server cookie from the software update service containing an identifier for the a target group of client computing devices *(i.e. For each software component, the update parameters database may include the identification of the specific network device or network devices in which the software component may be found, as well as the current version number of the software component. Furthermore, the information in the update parameters database may also include information pertaining to any update file for that software component, [0020])*; and

storing the server coolie at the client device such that the client device can use the server coolie to indicate membership in the target group of client computing devices *(i.e. For each software component, the update parameters*

database may include the identification of the specific network device or network devices in which the software component may be found, as well as the current version number of the software component. Furthermore, the information in the update parameters database may also include information pertaining to any update file for that software component, [0020]).

Gazdik, Lee do not specifically teach:

authenticating with an software update service prior to obtaining the software update information;

receiving an indication that software update service includes a server side authorization plug-in corresponding to a target group of client computing device;

determining that the computing device includes a client side authorization plug-in corresponding to the server side authorization plug-in;

sending an authorization cookie to the software update service in response to the determination, the authorization cookie identifying the target group of client computing devices; and

Colvin teaches:

authenticating with an software update service prior to obtaining the software update information *(i.e. If the current authorization interval has expired as indicated at 386, an attempt is made to connect to the server of the administrator as indicated at 388. If the connection is successful as determined by block 390, information including a licensed file update with a new authorization interval and/or various other information may be downloaded to the user or user computer as indicated at 396. Otherwise, an error message is*

displayed as indicated at 392 and the process terminates as represented by block 394, col. 16, lines 23-32);

receiving an indication that software update service includes a server side authorization plug-in corresponding to a target group of client computing device *(i.e. each code is supplied in the form of a plug-in module. Some codes may not be used in the automatic electronic update process but only in the manual password entry mode according to the present invention. This allow the monitor module to display the generated code which can be manually provided to an authorized software representative to obtain an authorization code for those users/computers unable or unwilling to electronically transfer authorization information as described in greater detail below, col. 12, line 41 to col. 13, line 19)*;

determining that the computing device includes a client side authorization plug-in corresponding to the server side authorization plug-in *(i.e. each code is supplied in the form of a plug-in module. Some codes may not be used in the automatic electronic update process but only in the manual password entry mode according to the present invention. This allow the monitor module to display the generated code which can be manually provided to an authorized software representative to obtain an authorization code for those users/computers unable or unwilling to electronically transfer authorization information as described in greater detail below, col. 12, line 41 to col. 13, line 19)*;

sending an authorization cookie to the software update service in response to the determination, the authorization cookie identifying the target

group of client computing devices *(i.e. To obtain a password update, registration code 250 is transferred to the authorized software representative (manually or automatically). The corresponding decoding table 268 is accessed to provide corresponding codes 252'-266'. In addition, one of the plurality of passwords associated with the password code is selected and supplied to the user or user computer as indicated generally by reference numeral 276. For continued authorization and operation of the protected software, the registration code and corresponding codes for the hardware, time, date, update number, etc. must match in addition to the balance of the password based on the serial number and other registration information stored in the database maintained by the authorized software representative, col. 13, lines 20-32).*

It would have been obvious to one of ordinary skill of the art having the teaching of Gazdik, Lee, and Colvin at the time the invention was made to modify the system of Gazdik, Lee to include the limitations as taught by Colvin. One of ordinary skill in the art would be motivated to make this combination in order to download a licensed file update with a new authorization interval in view of Colvin (col. 16, lines 23-32), as doing so would give the added benefit of monitoring computer software installed on a plurality of computers in communication with one another or a central computer to form a computer network include associating an activation code or password with the computer software for authorizing one or more copies of the software to be installed on computers associated with the computer network as taught by Colvin (col. 3, lines 21-39).

As per claim 13, Lee, as combined, teaches obtaining software update information to be installed on the computing device comprises determining that the software update is targeted to the target group of client computing devices *(i.e. There is also included a software update engine configured to send individual sets of the update parameters to individual ones of the plurality of local update agents at the plurality of network devices, wherein a first one of the plurality of local update agents disposed at a first one of the plurality of networked devices is configured to obtain, upon receiving a first one of the individual sets of the update parameters, a first update file using the first one of the individual sets of update parameters. The first update file represents a file containing data for updating a first one of the plurality of software components disposed at the first one of the plurality of networked devices, [0010])*.

As per claim 17, Colvin, as combined, teaches the method as recited in claim 11, further comprising:

authenticating with an software update service prior to obtaining the software update information *(i.e. If the current authorization interval has expired as indicated at 386, an attempt is made to connect to the server of the administrator as indicated at 388. If the connection is successful as determined by block 390, information including a licensed file update with a new authorization interval and/or various other information may be downloaded to the user or user computer as indicated at 396. Otherwise, an error message is*

displayed as indicated at 392 and the process terminates as represented by block 394, col. 16, lines 23-32), authentication including:

sending a configuration request message to the software update service *(i.e. If the current authorization interval has expired as indicated at 386, an attempt is made to connect to the server of the administrator as indicated at 388. If the connection is successful as determined by block 390, information including a licensed file update with a new authorization interval and/or various other information may be downloaded to the user or user computer as indicated at 396. Otherwise, an error message is displayed as indicated at 392 and the process terminates as represented by block 394, col. 16, lines 23-32)*;

receiving an indication that software update service includes a server side authorization plug-in corresponding to a target group of client computing device *(i.e. each code is supplied in the form of a plug-in module. Some codes may not be used in the automatic electronic update process but only in the manual password entry mode according to the present invention. This allow the monitor module to display the generated code which can be manually provided to an authorized software representative to obtain an authorization code for those users/computers unable or unwilling to electronically transfer authorization information as described in greater detail below, col. 12, line 41 to col. 13, line 19)*;

determining that the computing device includes a client side authorization plug-in corresponding to the server side authorization plug-in *(i.e. each code is supplied in the form of a plug-in module. Some codes may not be used in the*

automatic electronic update process but only in the manual password entry mode according to the present invention. This allow the monitor module to display the generated code which can be manually provided to an authorized software representative to obtain an authorization code for those users/computers unable or unwilling to electronically transfer authorization information as described in greater detail below, col. 12, line 41 to col. 13, line 19);

sending an authorization cookie to the software update service in response to the determination, the authorization cookie identifying the target group of client computing devices *(i.e. To obtain a password update, registration code 250 is transferred to the authorized software representative (manually or automatically). The corresponding decoding table 268 is accessed to provide corresponding codes 252'-266'. In addition, one of the plurality of passwords associated with the password code is selected and supplied to the user or user computer as indicated generally by reference numeral 276. For continued authorization and operation of the protected software, the registration code and corresponding codes for the hardware, time, date, update number, etc. must match in addition to the balance of the password based on the serial number and other registration information stored in the database maintained by the authorized software representative, col. 13, lines 20-32)*; and

receiving a server cookie from the software update service containing an identifier for the a target group of client computing devices *(i.e. To obtain a password update, registration code 250 is transferred to the authorized software representative (manually or automatically). The corresponding decoding table*

268 is accessed to provide corresponding codes 252'-266'. In addition, one of the plurality of passwords associated with the password code is selected and supplied to the user or user computer as indicated generally by reference numeral 276. For continued authorization and operation of the protected software, the registration code and corresponding codes for the hardware, time, date, update number, etc. must match in addition to the balance of the password based on the serial number and other registration information stored in the database maintained by the authorized software representative, col. 13, lines 20-32); and

storing the server coolie at the client device such that the client device can use the server coolie to indicate membership in the target group of client computing devices *(i.e. To obtain a password update, registration code 250 is transferred to the authorized software representative (manually or automatically). The corresponding decoding table 268 is accessed to provide corresponding codes 252'-266'. In addition, one of the plurality of passwords associated with the password code is selected and supplied to the user or user computer as indicated generally by reference numeral 276. For continued authorization and operation of the protected software, the registration code and corresponding codes for the hardware, time, date, update number, etc. must match in addition to the balance of the password based on the serial number and other registration information stored in the database maintained by the authorized software representative, col. 13, lines 20-32)*.

As per claim 18, Lee, as combined, teaches the method as recited in claim 17, wherein obtaining software update information to be installed on the computing device comprises determining that the software update is targeted to the target group of client computing devices *(i.e. There is also included a software update engine configured to send individual sets of the update parameters to individual ones of the plurality of local update agents at the plurality of network devices, wherein a first one of the plurality of local update agents disposed at a first one of the plurality of networked devices is configured to obtain, upon receiving a first one of the individual sets of the update parameters, a first update file using the first one of the individual sets of update parameters. The first update file represents a file containing data for updating a first one of the plurality of software components disposed at the first one of the plurality of networked devices, [0010])*.

As per claim 19, Lee, as combined, teaches the method as recited in claim 11, further comprising:

synchronizing available updates with a software update service, including traversing a software update hierarchy to identify specific software updates that apply to the computing device, the software update hierarchy including at least a base set of one or more base updates and a second set of one or more second tier updates *(i.e. The update schedule specifies the time when an update for a particular software component in a particular networked device should be performed. Optionally, the update schedule may also include a priority*

classification for the update. When the scheduled time arrives to update a particular software component on a particular networked device, a software update engine (which may include one or more individual sub-engines) sends the update parameters regarding the update file, along with any other parameters relevant to the update, to a local update agent local to the particular networked device on which the software component to be updated is located. The information sent includes, for example, parameters indicating where in the network or on the Internet the actual update file may be found and downloaded, [0022]), wherein

each base update includes one or more applicability condition to be tested at the computing device to determine if the base update is applicable to the computing device *(i.e. There is also included a software update engine configured to send individual sets of the update parameters to individual ones of the plurality of local update agents at the plurality of network devices, wherein a first one of the plurality of local update agents disposed at a first one of the plurality of networked devices is configured to obtain, upon receiving a first one of the individual sets of the update parameters, a first update file using the first one of the individual sets of update parameters. The first update file represents a file containing data for updating a first one of the plurality of software components disposed at the first one of the plurality of networked devices, [0010])*; and

each second tier update includes: (a) one or more applicability conditions to be tested at the computing device to determine if the second tier update is applicable to the computing device, (b) prerequisites identifying one or more

base updates that are required for proper installation of the second tier update, and (c) an indicator of whether or not the second tier update is a leaf update *(i.e. For each software component, the update parameters database may include the identification of the specific network device or network devices in which the software component may be found, as well as the current version number of the software component. Furthermore, the information in the update parameters database may also include information pertaining to any update file for that software component, [0020])*.

As per claim 20, Lee, as combined, teaches the method as recited in claim 19, wherein obtaining software update information to be installed on the computing device comprises receiving a selection of one or more software updates form among the identified specific software updates that apply to the computing device *(i.e. As another example, if the installation tasks were handled from a central location (e.g., from the centralized software update engine), the processing burden associated with processing a large number of updates using a single processing engine would have degraded performance. In contrast, the invention takes advantage of the distributed bandwidth in the network links and routers to allow the different networked device to more rapidly obtain their own required update files. Furthermore, the invention takes advantage of the distributed processing power in the networked devices to share the processing burden associated with updates, [0028])*.

Response to Arguments

Applicant's arguments with respect to claims 1-20 have been considered but are moot in view of the new ground(s) of rejection.

Applicant's arguments filed 01/15/09 have been fully considered but they are not persuasive. In page 1 of Remarks, Applicants states that "The invention is generally directed to controlling installation update behaviors on a client computer". However, the claim language only refers to "a computing device", but not "a client computer".

Conclusion

The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Miranda Le whose telephone number is (571) 272-4112. The examiner can normally be reached on Monday through Friday from 10:00 AM to 6:00 PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, James K. Trujillo, can be reached at (571) 272-3677. The fax number to this Art Unit is (571)-273-8300.

Any inquiry of a general nature or relating to the status of this application should be directed to the Group receptionist whose telephone number is (571) 272-2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/Miranda Le/
Primary Examiner, Art Unit 2169